AD-A033 433 STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB F/G 12/2
ROBUST IMPLEMENTATION OF LEMKE'S METHOD FOR THE LINEAR COMPLEME--ETC(U)
SEP 76 J A TOMLIN N00014-75-C-0865
UNCLASSIFIED SOL-76-24 NL

END

DATE FILMED 2-77

ROBUST IMPLEMENTATION OF LEMKE'S METHOD FOR THE LINEAR COMPLEMENTARITY PROBLEM

BY

J. A. TOMLIN

TECHNICAL REPORT SOL 76-24

SEPTEMBER 1976

Systems Optimization Laboratory

Department of Operations Research

Stanford University

Stanford California 94305

ROBUST IMPLEMENTATION OF LEMKE'S METHOD FOR THE LINEAR COMPLEMENTARITY PROBLEM

by

J.A. Tomlin

TECHNICAL REPORT SOL 76-24

September 1976

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH

Stanford University
Stanford, California

Research and reproduction of this report were partially supported by the U.S. Army Research Office - Durham, Contract DAAC-29-74-C-0034; the Office of Naval Research Contract N00014-75-C-0865; the National Science Foundation Grant DCR75-04544; and the U.S. Energy Research and Development Administration Contract E(04-3)-326 PA #18.

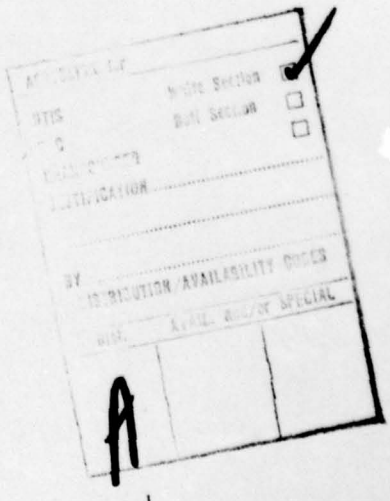
ROBUST IMPLEMENTATION OF LEMKE'S METHOD FOR THE LINEAR COMPLEMENTARITY PROBLEM

by

J.A. Tomlin

Abstract

This note discusses techniques for implementing Lemke's algorithm for the linear complementarity problem in a numerically robust way as well as a method for recovering from loss of feasibility or singularity of the basis. This recovery method is valid for both positive semi-definite M matrices and those with positive principal minors. It also allows a user to start from an advanced basis for such problems.



Introduction

While Lemke's algorithm [4] for the linear complementarity problem (LCP) is well established in the theoretical sense and to some extent as a computational procedure, it has not come into universal use for solving LCP's in the same way that the simplex method has for linear programs. This is due to several reasons. To begin with, some LCP's with special structure can be solved more efficiently by other special purpose algorithms (see [1]). It is also apparent that some problems (such as quadratic programs with a relatively small proportion of quadratic variables), which can be set up as LCP's and solved by Lemke's method, might be more efficiently solved by direct optimization methods. The question of applicability of the algorithm will not be considered here however. The problem we shall be concerned with is the stability of Lemke's algorithm in the face of numerical error and its ability to recover in the face of these errors.

The revised simplex method (see e.g., [5]), when it breaks down, usually does so in one of two ways -- a loss of feasibility in the solution, or the current "basis" is detected to be singular (or nearly so) when an attempt is made to invert it. Both of these situations are quite easily remedied. In the first case the algorithm can revert to Phase I of the simplex method to restore feasibility. In the second case dependent columns can be removed from the basis and replaced by logical (unit) vectors to give a new non-singular basis. The resulting basic solution will now almost certainly be

"Revised" Form of Lemke's Algorithm

Familiarity with both the Lemke algorithm and the revised simplex method is assumed, but we make the following observations for the sake of clarity.

The linear complementarity problem is customarily expressed in the form (see e.g., Cottle and Dantzig [2]):

Find w and z such that

$$w = q + Mz \tag{1}$$

$$w, z \geq 0 , \quad w^T z = 0$$

A new non-negative vector e which is positive in each component corresponding to a negative q_i, and a new variable z_0, are added to the problem, which can be stored internally in the form:

$$Iw - Mz - ez_0 = q . \tag{2}$$

Since column e is immediately introduced into the basis to provide a non-negative basic solution, and remains there until termination, it is important that e only contain the minimum number of elements required, otherwise the basis factorization or inverse representation may become extremely dense.

infeasible and Phase I must be carried out to restore feasibility. It is clear that the ability of the simplex method to recover in this way is a result of its two phase nature and the admissibility of any basis as a starting point for recovery. These features also enable the user to start the simplex method from any purported basis.

The Lemke algorithm is also vulnerable to loss of feasibility and singularity of the basis (we assume a revised simplex-like implementation). However since there is no "Phase I" and the basis is required to be of a special nature (an almost complementary set of columns) it is not immediately clear how these misfortunes can be overcome. Similarly we cannot, in general, start with a given basis unless it happens to be feasible and almost complementary. These are severe restrictions for practical computation for all but the smallest and simplest problems. In view of these difficulties, three new algorithms for LCP's were developed by Rarick [6] which did allow for a two-phase procedure and which successfully solved problems with hundreds of variables. At present, only two of these algorithms have been proved to converge and then only for semi-definite matrices. Since Lemke's algorithm is known to solve (theoretically) a wider class of problems, it seems worthwhile to try to find robust procedures for implementing it and to devise methods for dealing with infeasibility and basis singularity.

Let B be the current almost complementary feasible basis for (2) at any iteration and suppose (w_p, z_p) is the complementary pair of variables of which one has just become non-basic. The steps of an iteration are then extremely simple:

1. If z_0 has become non-basic or $z_0 < \epsilon$, for some small ϵ, terminate.

2. Determine which of w_p or z_p is to enter the basis (the complement of the variable which has just left the basis) and compute.

$$\alpha = B^{-1}u_p \quad \text{or} \quad \alpha = -B^{-1}m_p$$

where m_p is the p^{th} column of M and u_p the p^{th} unit vector.

3. Compute

$$\frac{\beta_r}{\alpha_r} = \min_{\alpha_i > 0} \frac{\beta_i}{\alpha_i} \qquad \text{where } \beta = B^{-1}q$$

4. Record the new non-basic pair and update β and the representation of B^{-1}. (Either the standard product form or some decomposition of B). Go to 1.

The basis is of course periodically reinverted or refactorized.

Stable Implementation

Unlike the simplex method, the variable to enter the basis is uniquely determined by the previous iteration. There is, in general, no possibility of rejecting the column because of unsatisfactory pivot size and using some other column. Because of this we must be as careful as possible in choosing the pivot in step 3 of each iteration. The first, and most obvious precaution is to use Harris's pivot row selection technique [3] to choose the largest available pivot in the event that there is a choice of pivots which maintain feasibility within a given tolerance ϵ_0. Briefly restated, this involves a two stage process:

3a) $$\varphi = \min_{\alpha_i \geq \epsilon_P} \frac{\beta_i + \epsilon_0}{\alpha_i}$$

3b) Choose β_r/α_r such that

$$\alpha_r = \max\{\alpha_i | \alpha_i \geq \epsilon_P, \frac{\beta_i}{\alpha_i} \leq \varphi\}$$

where ϵ_P is an absolute pivot tolerance.

As a further check we may impose a relative pivot tolerance test on α_r. Let $\alpha_{max} = \max|\alpha_i|$ and ϵ_R be a tolerance. Then if $\alpha_r \geq \epsilon_R \cdot \alpha_{max}$ the pivot is accepted. If not, the choice is suspended and the basis is reinverted so that α will be recomputed as accurately as possible (or practical). If the recomputed α_r again fails the relative pivot test it must now be accepted, but

with greater confidence. Clearly if ϵ_R is too small the test is ineffective and if it is too large reinversion will be carried out more frequently than necessary. We have found a value of 10^{-8} to be satisfactory, when the relative machine precision is about 10^{-15}.

Much of the concern about accuracy can be removed by using one of the stable techniques for updating triangular factors of the basis now available, for example Saunders' efficient version [7] of the Bartels-Golub method. We should also point out that use of a triangular updating technique is particularly beneficial on sparsity grounds, since the e column in (2) persists in the basis and may be dense. In this case we expect it to be assigned as one of the right-most columns of U in the LU decomposition of B produced by almost any sparsity-preserving technique. In this position it will have minimal effect on the build up of non-zeros during the updating process.

As a final measure we may periodically check the relative error in β by performing one iteration of iterative refinement with the current representation of B^{-1}. If this relative error is unsatisfactory reinversion may be called for.

Matrix Scaling

If the matrix M is badly scaled or ill-conditioned the possibility of numerical difficulties will be magnified. It therefore seems worthwhile to have scaling facilities available. Again the situation here is not as straight-forward as it is for the simplex

method, as it is not clear that all matrices M and right hand sides q for which the LCP is solvable by Lemke's algorithm preserve this property when the problem is scaled. Fortunately this solvability is preserved for two large classes of matrices -- those which are positive semi-definite and matrices with positive principal minors. If we let $R = \text{diag}(r_1, \ldots, r_n)$ and $C = \text{diag}(c_1, \ldots, c_n)$ be the diagonal matrices of positive row and column scales, then it is clear that if

$$M' = RMC$$

then M' retains semi-definiteness and/or positive principal minors if M has these properties. Hence, the scaled problem

$$w' = q' + M'z'$$

$$w', z' \geq 0 \qquad (w')^T z' = 0$$

where $w' = Rw$, $q' = Rq$ and $z' = C^{-1}z$ remains solvable.

There remains some ambiguity however, about the applicability of scaling to the more esoteric classes of matrices for which Lemke's algorithm is known to be effective. In particular, it is not clear whether the additional column e must be row scaled in the same way as the matrix, or constructed after scaling. Further research on this question is called for.

The choice of scaling methods for M is essentially the same as for linear programming. An earlier study of LP scaling [8] indicates that the method due to Curtis and Ried is very effective and we recommend it here.

Recovery Procedure

Despite all precautions there are bound to be occasions when the algorithm breaks down, leading to an infeasible solution or a singular "basis." Rather more frequently, a user may wish to supply a basis which turns out to be infeasible or even singular but is close in some sense to the solution. Both of these difficulties can be overcome, at least for the two large classes of M with which we are principally concerned -- positive semi-definite and positive principal minors.

As soon as either error condition is detected, the e column is dropped from the basis and replaced by the unit column u_p corresponding to the w_p variable in the non-basic complementary pair. If the resulting basis remains singular, dependent columns m_j are dropped from the basis and replaced by their complementary unit columns u_j. This process must lead to a non-singular complementary basis -- that is one containing a (possibly empty) set of columns $-M_J$ and a complementary set of columns of the unit matrix $I_{\bar{J}}$. Let $\bar{B}$ be the resulting basis.

Now consider the transformed problem

$$\bar{w} = \bar{q} + \bar{M}\bar{z}$$

$$\bar{w},\ \bar{z}' \geq 0 \qquad \bar{w}^T\bar{z} = 0 \tag{3}$$

where $\bar{w} = \bar{B}^{-1}w$, $\bar{q} = \bar{B}^{-1}q$ $\bar{M} = \bar{B}^{-1}M$ and $\bar{w}$, $\bar{z}$ are a complementary permutation of w, z.

By the definition of $\bar{B}$, $\bar{M}$ is a principal transform of M and if M is positive semi-definite and/or has positive principal minors so does $\bar{M}$ (see Cottle and Dantzig [2] Theorems 8 and 9). Thus, if the original problem (1) is solvable by Lemke's algorithm, so is the transformed problem (3). All that remains is to construct a new column $\bar{e}$ with positive elements corresponding to the negative elements of $\bar{q}$. This is then assigned a new variable $\bar{z}_0$ and added to the problem, the pivot row (and new nonbasic pair) being chosen to give a new almost complementary feasible solution.

It is important to note that we continue in practice to deal with the original M and q and $\bar{B}$ not $\bar{M}$ and $\bar{q}$, hence we require $\bar{e}$ also in terms of the unit basis. This is of course available as $e' = \bar{B}\bar{e}$, and e is overwritten with e' in the interval representation of the problem (2).

To illustrate the fallibility of this recovery procedure consider the small example:

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -1 \\ -5 \end{bmatrix} + \begin{bmatrix} -1 & 4 \\ -3 & 11 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

Constructing $e = (1, 1)$ and applying Lemke's algorithm leads in two iterations to the correct solution $w_1 = 9/11$, $w_2 = 0$, $z_1 = 0$, $z_2 = 5/11$. If however, we pivot on m_{11} to produce the principal transform:

$$\begin{bmatrix} z_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \end{bmatrix} + \begin{bmatrix} -1 & 4 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} w_1 \\ z_2 \end{bmatrix}$$

the resulting M matrix is the negative of a Minkowski matrix and the algorithm terminates after the first iteration in the almost complementary ray $z_0 = 2 + \theta$, $z_1 = 1 + 5\theta$, $z_2 = \theta$, $w_1 = w_2 = 0$.

What recovery procedures (if any) are applicable to problems which lie outside the classes we have considered therefore remains a topic for further research.

Conclusion

With the numerical techniques and the recovery procedure discussed in this note there is no reason that Lemke's method should not be used for a wide class of large scale linear complementarity problems. With the current exception of LU updating we have implemented these techniques in our code LCPL [9] and found it highly satisfactory. So far our experience has only been with problems of up to a little more than 500 variables; however, some of these had led to difficulties of the type described with other implementations.

Acknowledgements

The author is indebted to R.W. Cottle, B.C. Eaves, and M.A. Saunders for a number of helpful discussions.

References

[1] R.W. Cottle, "Computational Experience with Large Scale Linear Complementarity Problems," Technical Report SOL 74-13, Systems Optimization Laboratory, Stanford University. September 1974.

[2] R.W. Cottle and G.B. Dantzig, "Complementary Pivot Theory of Mathematical Programming," pp. 115-138 in Mathematics of the Decision Sciences, Part 1 (G.B. Dantzig and A.F. Veinott, Jr., eds.) American Mathematical Society, Providence R.I. (1968).

[3] P.M.J. Harris, "Pivot Selection Methods of the Devex LP Code," Mathematical Programming Study 4, pp. 30-57 (1975).

[4] C.E. Lemke, "Bimatrix Equilibrium Points and Mathematical Programming," Management Science 11, pp. 681-689 (1965).

[5] W. Orchard-Hays, Advanced Linear Programming Computing Techniques, McGraw-Hill, New York (1968).

[6] D.C. Rarick, "Solving Large Problems with Linear Constraints and Quadratic Objective Function," Presentation to the Joint ORSA/TIMS National Meeting, Philadelphia, March 31 - April 2 (1976).

[7] M.A. Saunders, "A Fast, Stable Implementation of the Simplex Method Using Bartels-Golub Updating," pp. 213-226 in Sparse Matrix Computations (J.R. Bunch and D.J. Rose, eds.) Academic Press, New York (1976).

[8] J.A. Tomlin, "On Scaling Linear Programming Problems," Mathematical Programming Study 4, pp. 146-166 (1975).

[9] J.A. Tomlin, "Users Guide to LCPL -- A Program for Solving Linear Complementarity Problems by Lemke's Method," Technical Report SOL 76-16, Systems Optimization Laboratory, Stanford University. August 1976.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: (14) SOL-76-24
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): (6) Robust Implementation of Lemke's Method for the Linear Complementarity Problem.
5. TYPE OF REPORT & PERIOD COVERED: (9) Technical Report.
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): (10) J.A. Tomlin
8. CONTRACT OR GRANT NUMBER(s): (15) N00014-75-C-0865, DAAC29-74-C-0034

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Operations Research, SOL
Stanford University
Stanford, CA 94305

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NR-047-064

11. CONTROLLING OFFICE NAME AND ADDRESS
Operations Research Program Code 434
Office of Naval Research
Arlington, VA 22217

Mathematics Division
U.S. Army Research Office
Box CM, Duke Station
Durham, N.C. 27706

(12) 17p.

(11) 12. REPORT DATE: Sept[illegible] 1976
13. NUMBER OF PAGES: 15
14. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Mathematical Programming
Linear Complementarity
Sparse Matrices

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This note discusses techniques for implementing Lemke's algorithm for the linear complementarity problem in a numerically robust way as well as a method for recovering from loss of feasibility or singularity of the basis. This recovery method is valid for both positive semi-definite M matrices and those with positive principal minors. It also allows a user to start from an advanced basis for such problems.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 765

LB